

UNITED STATES DISTRICT COURT  
DISTRICT OF MASSACHUSETTS

DATATERN, INC.

Plaintiff,

v.

BLAZENT, INC., MICROSTRATEGY INC.,  
CARL WARREN AND COMPANY  
INCORPORATED, LANCET SOFTWARE  
DEVELOPMENT, INC., AIRLINES  
REPORTING CORP., MAGIC SOFTWARE  
ENTERPRISES LTD., MAGIC SOFTWARE  
ENTERPRISES, INC., TERADATA  
CORPORATION, INFORMATICA  
CORPORATION, EPICOR SOFTWARE  
CORPORATION, and PREMIER, INC.

Defendants

1:11-cv-11970-FDS

(Consolidated)

---

**MEMORANDUM IN SUPPORT OF DEFENDANT MICROSTRATEGY  
INCORPORATED'S MOTION FOR SUMMARY JUDGMENT OF NON-  
INFRINGEMENT AND ATTORNEY'S FEES**

---

## Table of Contents

I.	Introduction.....	2
II.	Summary of argument.....	5
III.	Procedural History .....	7
	A.    DataTern's Massachusetts Litigation.....	7
	B.    DataTern's New York Litigation .....	10
	C.    DataTern's Stipulation of Non-Infringement.....	11
IV.	Technical Background .....	12
	A.    Technology Tutorial.....	12
	1.    Object-Oriented Programming.....	12
	2.    Relational Databases .....	15
	B.    The Disclosure of the '502 Patent.....	16
	C.    Claim 1 of the '502 Patent .....	18
	D.    The New York Claim Constructions and Summary Judgment Order.....	20
	E.    DataTern's Infringement Contentions .....	21
	1.    “object model” .....	21
	2.    “create at least one an interface object” .....	22
	3.    “associated with an object corresponding to a class associated with the object oriented software application” .....	22
	4.    “runtime engine” .....	23
	5.    “interface object . . . to access data from the relational database.....	23
V.	The Court Should Grant Summary Judgment Of Noninfringement.....	24
	A.    Legal Standards.....	24
	1.    Summary Judgment .....	24
	2.    Collateral Estoppel Effect of Claim Construction Rulings .....	25
	B.    The Court's Should Grant Summary Judgment of Noninfringement Because DataTern Concedes that the Accused Products Do Not Create an Interface Object.....	26

C. The Court Should Grant Summary Judgment on Two Further Grounds Mandated By the New York Court's Claim Constructions .....	27
1. DataTern Is Bound By Those of the New York Court's Claim Constructions That Were Essential to Its Judgment .....	27
2. The Accused Products Do Not Satisfy the "Object Model" Limitation .....	28
3. The Accused Products Do Not Satisfy the "Runtime Engine" Limitation.....	29
D. The Court Should Further Grant Summary Judgment on Three Additional Grounds, Because DataTern's Infringement Contentions Do Not Establish Infringement.....	29
1. Summary Judgment Is Appropriate in Light of DataTern's Infringement Contentions .....	29
2. DataTern Cannot Show That the Accused Products Create an Interface Object "Associated With An Object Corresponding To A Class Associated With The Object Oriented Software Application".....	31
3. DataTern Cannot Show that MicroStrategy's Products Include a "Code Generator" .....	32
4. DataTern Cannot Show that MicroStrategy's Products Invoke an Interface Object "With the Object Oriented Application" .....	33
VI. The Court Should Declare This To Be An Exceptional Case And Award Microstrategy Its Reasonable Attorneys' Fees .....	33
A. Legal Standards.....	33
B. The Court Should Declare This to Be an Exceptional Case and Award MicroStrategy Its Reasonable Attorneys' Fees for Defending Against DataTern's Baseless Allegations .....	34
1. DataTern's Allegations That Microstrategy's Products Satisfied The "Object Model" Limitation Are Baseless.....	34
2. DataTern Lacks Any Basis for its Allegations that the Accused Products Create an Interface Object "Associated With An Object Corresponding To A Class Associated With The Object Oriented Software Application" .....	35
3. DataTern's Allegations that the Accused Products Invoke an Interface Object "With the Object Oriented Application" Are Baseless .....	36
4. DataTern's Allegations that the Accused Products Include a "Code Generator" Are Baseless .....	36

C. MicroStrategy Should at Least Be Awarded Attorney's Fees Resulting from DataTern's Vexatious Litigation Tactics.....	37
1. DataTern's Lawsuit Against MicroStrategy's Customers Was Motivated By an Improper Purpose.....	37
2. DataTern Unreasonably Forced MicroStrategy to Collect and Make Available Source Code DataTern Never Intended to Review .....	38
VII. CONCLUSION.....	40

## TABLE OF AUTHORITIES

### CASES

<i>Biovail Laboratories Intern. v. Int爾genx Corp.</i> , Civ. No. 09-605-LPS, 2010 WL 5625746 (D. Del. December 27, 2010) .....	25
<i>DataTern v. Blazent</i> , Civ. No. 1:11-cv-11970 .....	7
<i>DataTern v. InQuira</i> , Civ. No. 1:11-cv-12027 .....	7
<i>Eon-Net LP v. Flagstar Bancorp</i> , 653 F.3d 1314 (Fed. Cir. 2011).....	37
<i>Highmark, Inc. v. Allcare Health Mgmt. Sys.</i> , 687 F.3d 1300 (Fed. Cir. 2012).....	34
<i>MarcTec, LLC v. Johnson &amp; Johnson</i> , 664 F. 3d 907 (Fed. Cir. 2012).....	33, 34, 35, 37
<i>Molinaro v. Fannon/Courier Corp.</i> , 745 F.2d 651 (Fed. Cir. 1984).....	25, 27
<i>Monsanto Co. v. Syngenta Seeds, Inc.</i> , 503 F.3d 1352 (Fed. Cir. 2007).....	33
<i>Moore U.S.A., Inc. v. Standard Register Co.</i> , 229 F.3d 1091 (Fed. Cir. 2000).....	32
<i>Nomadix, Inc. v. Hewlett-Packard Co.</i> , 2012 WL 682874 (C.D. Cal. Mar. 1, 2012) .....	29, 30, 32
<i>O2 Micro Int'l Ltd. v. Monolithic Power Sys. Inc.</i> , 467 F.3d 1355 (Fed. Cir. 2005).....	30
<i>Old Reliable Wholesale, Inc. v. Cornell Corp.</i> , 635 F.3d 539 (Fed. Cir. 2011).....	39
<i>Pfaff v. Wells Electronics, Inc.</i> , 5 F.3d 514 (Fed. Cir. 1993).....	25, 28
<i>Regents of Univ. of Cal. v. Dako N. Am., Inc.</i> , No. C 05-03955, 2009 WL 1083446 (N.D. Cal. Apr. 22, 2009) .....	25
<i>Ricoh Co., Ltd. v. Aeroflex Inc.</i> , 279 F.Supp.2d 554 (D. Del.2003) .....	37
<i>Simas v. First Citizens' Fed. Credit Union</i> , 170 F.3d 37 (1st Cir. 1999) .....	25, 30

**STATUTES**

35 U.S.C. § 285.....	4, 33, 37
<i>35 U.S.C. S 285 .....</i>	4

**OTHER AUTHORITIES**

Christopher A. Harkins, <i>Fending Off Paper Patents and Patent Trolls: A Novel "Cold Fusion" Defense Because Changing Times Demand It</i> , 17 Alb. L.J. Sci. & Tech. 407, 444 (2007).....	4
Fed. R. Civ. P. 54.....	39
Fed. R. Civ. P. 54(d)(2)(B)(iii) .....	34, 35, 39
Fed. R. Civ. P. 56(d) .....	25, 30

## I. INTRODUCTION

There is no dispute that judgment should enter against DataTern. But DataTern's proposed judgment sets out only one of several grounds on which MicroStrategy is entitled to judgment. And DataTern's proposed judgment fails to include a finding that this is an exceptional case, a prerequisite to awarding attorneys' fees, to which MicroStrategy is entitled. As a result, MicroStrategy requests that the Court enter its proposed judgment, filed herewith, and that the Court set a schedule for the determination of the value of MicroStrategy's reasonable attorneys' fees.

With respect to MicroStrategy's request for its reasonable attorneys' fees, this truly is a case in which an award of fees is warranted. DataTern's litigation has been objectively baseless from the beginning. Early indications that even DataTern was aware of the tenuous nature of this litigation came when:

- In November 2011, DataTern filed seventeen lawsuits against MicroStrategy customers and business partners, abruptly dismissed eleven that were not assigned to Judge Stearns, then filed five more, in an apparent effort to exert pressure on MicroStrategy by suing an array of customers with few resources to defend themselves;
- During the parties' Rule 26(f) conference, DataTern refused to provide infringement contentions on the ordinary schedule set out in the local rules, claiming that it needed to review the defendants' source code before it would disclose its Rule 11 basis for filing this case; and
- Also during the parties' Rule 26(f) conference, DataTern refused to state the theory under which DataTern was suing the non-MicroStrategy defendants, and further refused to stay the cases against them in favor of the MicroStrategy case, again seeking to exert pressure against MicroStrategy by increasing the litigation costs of its customers without having any independent basis for including them in the litigation.

As the case progressed, the lack of basis for DataTern's allegations became even more apparent. In April 2011, DataTern served infringement contentions that failed to disclose a legally sufficient basis for its infringement theory. When ordered to supplement its

contentions, DataTern served a second set of contentions that fared no better. When challenged, DataTern responded that it needed to see MicroStrategy's source code in order to provide viable contentions. But when offered the opportunity to review MicroStrategy's source code, DataTern refused to inspect it, suddenly claiming that it was not obligated to provide revised contentions that took into account the source code it had previously been demanding to review.

Any foundation for DataTern's infringement theories was further undermined in August 2012, when the United States District Court for the Southern District of New York issued its claim construction order in separate case involving the patent-in-suit. But rather than concede noninfringement on each of the grounds mandated by the New York court's orders, DataTern has only agreed to stipulate to judgment on one narrow term. The result of DataTern's approach, if accepted, would be that if the New York court's claim construction is affirmed in less than its entirety, this case could be remanded for further unnecessary proceedings.

This Court should enter judgment of noninfringement on each of the grounds to which the defendants are entitled.<sup>1</sup> DataTern's infringement theories have been deficient from the beginning. It would be unfairly prejudicial to require the defendants to now wait for the Federal Circuit to resolve the appeal in the New York case, only to risk returning to this court if the Federal Circuit affirms the New York court's constructions on some ground other than the one that DataTern concedes here. The more efficient and just approach is for this Court to effect a complete resolution of this case to avoid the possibility of an unnecessary remand.

---

<sup>1</sup> MicroStrategy has bases for noninfringement beyond what is set out in this brief, but most of these other bases would require this Court to conduct further claim construction proceedings. As a result, MicroStrategy focuses in this motion on terms that either need no construction or were construed and relied on by the New York court in granting summary judgment of noninfringement there.

This Court should also declare this to be an “exceptional case” under 35 U.S.C. § 285 and grant MicroStrategy its reasonable attorneys’ fees. Despite this Court’s order that DataTern provide infringement contentions that set forth its Rule 11 basis for filing this lawsuit, DataTern’s infringement contentions remained deficient, as explained in detail below. Where, as here, a plaintiff has continued with unjustified assertions of patent infringement, particularly when coupled with improper litigation tactics and vexatious conduct, a declaration of exceptional case and an award of attorneys’ fees is appropriate.

Non-practicing entities asserting questionable theories of patent infringement pose a systemic problem with respect to the country’s patent litigation regime. In our system,

manufacturing defendants [are] susceptible to being victimized by scare-and-run litigation tactics that infect its business environment and may lower its stock prices if it takes a chance on trial instead of an early settlement. The uncertain, insecure, and precarious cloud over the manufacturer’s products renders it particularly helpless and immobile so long as there is a disparity of settlement conditions: A manufacturer making decisions based strictly on cost analysis while an overly optimistic patent troll bets on a game theory expecting to be “bought off” based on the higher relative cost of trial exposure to the defendant.

Christopher A. Harkins, *Fending Off Paper Patents and Patent Trolls: A Novel "Cold Fusion" Defense Because Changing Times Demand It*, 17 Alb. L.J. Sci. & Tech. 407, 444 (2007) (footnotes omitted). “[T]he result is a system that encourages aggressively seeking licenses and litigating questionable patents.” Daniel Kennedy, *Holding Parent Corporations Liable for Attorneys’ Fees Under 35 U.S.C. S 285 of the Patent Statute*, 61 Baylor L. Rev. 999, 999 (2009). Section 285’s fee-shifting provision is an important tool in a defendant’s arsenal. By shifting the cost of defending unmeritorious lawsuits, the provision alters the plaintiff’s “game theory” analysis (see Harkins at 444) and provides a disincentive to bringing unmeritorious or unfounded patent litigation in the first place. The Court should

apply the statute here, declare this to be an exceptional case, and grant MicroStrategy its reasonable attorneys' fees.

## II. SUMMARY OF ARGUMENT

1. Based on the New York court's claim constructions and summary judgment order, and the plain language of the claims, the Court should grant MicroStrategy summary judgment of non-infringement on six grounds:

**a. The accused products do not create an “interface object.”** The New York court construed this claim term to require “generat[ing] code for at least one class.” DataTern concedes that this requirement is not satisfied. There is no dispute that MicroStrategy is entitled to judgment on at least this ground.

**b. The accused products do not allow for the selection of an “object model.”** The New York court construed “object model” to require the inclusion of “object classes and inheritance relationships among classes.” The alleged “object model,” however, called MicroStrategy's Logical Data Model, is merely a mechanism for a customer to identify relationships between preexisting business data. It has nothing to do with object classes or inheritance relationships. As a matter of law, the Logical Data Model is not an “object model.”

**c. The accused products do not utilize a “runtime engine.”** The New York court construed the phrase “runtime engine” to mean, among other items, software that “the object oriented software application depends on to run.” Nothing in DataTern's infringement contentions suggests that the alleged “runtime engine” in MicroStrategy's software is required to run any alleged object oriented software application. Because DataTern cannot satisfy its summary judgment burden of proof on this element, the Court should grant summary judgment.

**d. The alleged interface object is not “associated with an object corresponding to a class associated with the object oriented software application.”** The claims of the '502 patent require interface objects that are associated with objects in an

object-oriented software application. DataTern’s infringement contentions simply fail to explain how this limitation is satisfied. DataTern cannot satisfy its summary judgment burden, and the Court should grant summary judgment of noninfringement on this ground as well.

**e. The accused products do not include a “code generator.”** Claim 10 of the ’502 patent expressly recites a “code generator” for creating interface objects. DataTern has already conceded that there is no code generator that creates interface objects. Summary judgment on this element is warranted as well.

**f. The accused products do not invoke an interface object “with the object oriented application.”** Claim 1 of the ’502 patent requires that an interface object be invoked “with the object oriented application” to access data in a database. DataTern ignores this requirement in its infringement contentions. Summary judgment on this ground is therefore also appropriate.

2. The Court should declare this to be an exceptional case and grant MicroStrategy its reasonable attorneys’ fees. DataTern never had a basis for bringing this lawsuit. DataTern could not have believed that at least the “object model,” “associated with an object corresponding to a class associated with the object oriented software application,” “with the object oriented application,” or “code generator” limitations were satisfied.

3. DataTern’s litigation conduct also supports a finding of exceptional case and a grant of attorneys’ fees. DataTern sued MicroStrategy’s customers with no basis, then unjustifiably refused to minimize costs by staying the cases against them, intending only to exert pressure on MicroStrategy. DataTern also unreasonably multiplied costs by repeatedly promising improved infringement contentions after it had had an opportunity to review MicroStrategy’s source code, yet then, once the source code was made available, refusing to inspect the code. MicroStrategy spent significant sums collecting its source code and

preparing it for production. The Court should declare this an exceptional case and require DataTern to reimburse MicroStrategy for its fees expended to collect and produce source code.

### **III. PROCEDURAL HISTORY**

#### **A. DataTern’s Massachusetts Litigation**

Between November 7, 2011 and November 15, 2011, DataTern filed seventeen lawsuits in this District alleging infringement of U.S. Patent No. 6,101,502 (the “502 patent”). (SOF<sup>2</sup> ¶ 1.) The originally-filed complaints made no mention of MicroStrategy. (See, e.g., Civ. No. 1:11-cv-12024, D.I. 1 at ¶ 10 (alleging infringement as a result of “Carl Warren’s ‘mycarlwarren.com’ as part of its claims and litigation management services”).)

On December 12, 2012, DataTern voluntarily dismissed nearly every case it filed other than the four cases assigned to Judge Stearns.<sup>3</sup> On December 13, 2012, in each of the remaining cases assigned to Judge Stearns, DataTern amended its complaint to assert infringement both as a result of the defendants’ independent actions (unrelated to their use of MicroStrategy products) as well as through their use of MicroStrategy products. (SOF ¶ 2.)

On December 14, 2011, DataTern filed a second round of five cases, this time marking them (on the Civil Cover Sheet) as related to Civ. No. 1:11-cv-12024—assigned to Judge Stearns. (SOF ¶ 3.) In violation of Local Rule 40.1(G)(2), DataTern did not list “each” earlier related case it had filed, the earliest of which was assigned to Judge Dein. (SOF ¶ 4.)

---

<sup>2</sup> “SOF” refers to MicroStrategy’s Rule 56.1 Statement of Undisputed Material Facts, filed herewith.

<sup>3</sup> DataTern dismissed Civ. Nos. 1:11-cv-11975-DJC, 1:11-cv-11976-LTS, 1:11-cv-11977-RBC, 1:11-cv-11980-DJC, 1:11-cv-11982-MLW, 1:11-cv-11983-DJC, 1:11-cv-11984-GAO, 1:11-cv-12022-GAO, 1:11-cv-12023-DJC, 1:11-cv-12029-JLT, and 1:11-cv-12030-DJC, assigned to Judges Casper, Sorokin, Collings, Wolf, O’Toole, and Tauro. DataTern did not dismiss any of the cases assigned to Judge Stearns: 1:11-cv-12024-RGS, 1:11-cv-12025-RGS, 1:11-cv-12026-RGS, and 1:11-cv-12028-RGS. DataTern also did not dismiss *DataTern v. Blazent*, Civ. No. 1:11-cv-11970, then assigned to Judge Dein, or *DataTern v. InQuira*, Civ. No. 1:11-cv-12027, assigned to Judge O’Toole.

Rather, DataTern listed only a single case—one assigned to Judge Stearns. (*Id.*) One of these cases was filed directly against MicroStrategy. (SOF ¶ 5.) The remainder asserted infringement against each of the defendants both individually and as a result of their use of MicroStrategy products. (SOF ¶ 6.)

To avoid unnecessary, duplicative litigation, and to reduce the costs on its customers, MicroStrategy moved to intervene and stay the cases filed against its customers and business partners. (SOF ¶ 7.) The Court consolidated the actions into a single case but denied MicroStrategy’s motion to stay. (SOF ¶ 8.)

In the parties’ Rule 16 Joint Statement, the defendants explained that they had asked (during their Rule 26(f) conference) whether DataTern intended to pursue claims against MicroStrategy’s customers “separate and independent of operation of the products with technology from MicroStrategy,” but that DataTern had not responded. (SOF ¶ 9.) Also in their joint statement, the parties agreed that the plaintiff’s infringement “disclosures must be supplemented within 90 days of production of a party’s source code to include specific citations to such source code for each claim element at issue.” (SOF ¶ 10.)

In its Scheduling Order, the Court ordered that DataTern would have “until June 1, 2012 to indicate to the court whether it intends to pursue claims of infringement against each non-MicroStrategy defendant for alleged infringement independent of each defendant’s use of MicroStrategy products (and if so identify which defendants and which products).” (SOF ¶ 11.) The Court also ordered that DataTern was required to serve infringement contentions by April 26, 2012. (SOF ¶ 12.)

On April 26, 2012, DataTern served its infringement contentions. In them, DataTern finally revealed that the “focus of this case has been and remains the MicroStrategy technology.” (SOF ¶ 13.) Thus, on May 10, 2012, the customer defendants moved to stay

DataTern's allegations against them. (SOF ¶ 14.) DataTern opposed the stay. (SOF ¶ 15.) The Court granted the stay over DataTern's opposition. (SOF ¶ 16.)

On May 7, 2012, MicroStrategy informed DataTern that its infringement contentions were deficient, and requested that DataTern serve a supplement. (SOF ¶ 17.) DataTern responded stating that it "sought production of the source code before providing its infringement contentions." (SOF ¶ 18.) Thus, DataTern "suggest[ed] that MicroStrategy provide its source code so that DataTern can supplement its infringement contentions." (SOF ¶ 19.) DataTern also pledged that "[c]onsistent with the Joint Statement Pursuant to Local Rules 16.1(D) and 16.6, nonetheless, DataTern intends to supplement its infringement contentions within 90 days of production of MicroStrategy's source code." (SOF ¶ 20.)

On May 25, 2012, MicroStrategy moved the Court to compel DataTern to provide sufficient contentions. (SOF ¶ 21.) DataTern's opposition relied on its lack of access to MicroStrategy's source code, describing the "necessity that review of the source code plays in confirming and defining the allegedly infringing products in software cases" such as this. (SOF ¶ 22.) Then, DataTern cited its reservation of rights to amend its contentions "since it did not have access to the source code." (SOF ¶ 23.)

In its July 2, 2012, order granting in part the defendants' motion, the Court agreed that DataTern's contentions were deficient and stated that "allegations of patent infringement must be supported by a good faith basis (under Rule 11)." (SOF ¶ 24.) The Court ordered DataTern to amend its infringement contentions and specifically identify "where each asserted claim element is found within each accused product or combination and supporting evidence." (SOF ¶ 25.)

DataTern served amended contentions on July 20, 2012. (SOF ¶ 26.) MicroStrategy informed DataTern that its amended infringement contentions were still deficient and failed to demonstrate DataTern's basis for bringing this case. (SOF ¶ 27.) DataTern replied that it

would be “glad to supplement after reviewing the source code when it is produced.” (SOF ¶ 28.)

MicroStrategy made source code available to DataTern on August 10, 2012. (SOF ¶ 29.) At the same time, MicroStrategy requested that DataTern—pursuant to the parties’ Joint Statement, the local rules, and DataTern’s previous pledge—amend its contentions to provide source code citations within 90 days. (SOF ¶ 30.) DataTern responded to MicroStrategy’s request stating, for the first time, “[W]e disagree with your contention that DataTern is required to inspect the MicroStrategy source code within 90 days.” (SOF ¶ 31.)

Soon thereafter, DataTern moved to stay the case in light of a claim construction order that had been issued by the United States District Court for the Southern District of New York. (SOF ¶ 32.) In response, MicroStrategy opposed the stay and requested that DataTern be held to its obligation to provide amended contentions with citations to MicroStrategy’s source code. (SOF ¶ 33.) The Court granted DataTern’s motion to stay, but noted that “[t]he stay shall not, however, prohibit DataTern from inspecting the software source code made available by MicroStrategy.” (SOF ¶ 34.) During the hearing on the motion, the Court advised:

“[W]hat I don’t want to do is to get to, let’s say, January and then have DataTern say we’ve got to go inspect the source code and that’s going to take 90 days or whatever. In other words, *if you think you need to [inspect MicroStrategy’s source code], the time to do it is now.*”

(SOF ¶ 35 (emphasis added).) DataTern has not, to date, inspected MicroStrategy’s source code.

## **B. DataTern’s New York Litigation**

Meanwhile, DataTern was litigating separate lawsuits involving the ’502 patent and another patent, U.S. Patent No. 5,937,402 (the “’402 patent”), in the Eastern District of Texas and in the Southern District of New York. (SOF ¶ 36.) Of particular relevance here, in April 2011, Microsoft Corp. and SAP AG filed declaratory judgment actions against DataTern in

the Southern District of New York requesting a judgment that the '402 and '502 patents were not infringed by Microsoft's or SAP's software, and that the claims of the patents were invalid. (*Id.*) On July 1, 2011, the court consolidated the cases. (*Id.*)

On August 24, 2012, the New York court entered an order construing certain terms of the '402 and '502 patents. (SOF ¶ 37.) On September 12, 2012, DataTern, conceding that it could not prove infringement under the New York court's claim constructions, filed a motion seeking the entry of a declaratory judgment of noninfringement. (SOF ¶ 38.) Ultimately, to ensure a complete record for the Federal Circuit, the plaintiffs elected to file partially-unopposed motions for summary judgment, which the court granted on December 19, 2012. (SOF ¶ 39.) Judgment entered against DataTern on December 26, 2012. (SOF ¶ 40.)

On January 9, 2013, SAP filed a motion for attorneys' fees. (SOF ¶ 41.) In relevant part, SAP argued that DataTern's infringement theory for "runtime engine" and "object model" lacked any legitimate basis. (SOF ¶ 42.) DataTern opposed (SOF ¶ 43.), and the Court has not yet ruled.

### **C. DataTern's Stipulation of Non-Infringement**

On January 3, 2013, in response to a motion filed by MicroStrategy requesting that the stay be lifted and that a scheduling order be entered, DataTern conceded that judgment in MicroStrategy's favor was appropriate:

DataTern agrees that in light of the New York Court's Final Judgments, the interests of judicial economy are best served by a judgment in a form and under terms agreed to by the parties to this action so that this matter may proceed on appeal in tandem with the New York action.

(SOF ¶ 44.) As a result, the Court ordered as follows:

What I'm going to do is to continue the stay in both cases, which are no[w] consolidated, give the parties 30 days in which to seek to negotiate an agreed form of judgment. . . . I suppose you could separately move for judgment if you disagree about what they look like or what they ought to cover. . . .

(January 4, 2013 Tr. at 6:23-7:4.) The deadline for the parties' proposed judgments was February 4, 2013. (*Id.* at 7:14.)

DataTern provided a draft proposed judgment to MicroStrategy on January 28, 2013. (SOF ¶ 47.) DataTern's stipulation acknowledged that "once the appropriate construction is determined by the Federal Circuit Court of Appeals it will have the effect of *stare decisis* with respect to the '502 Patent." (SOF ¶ 48). And DataTern's draft concedes that "if the NY Markman Order is applied to MicroStrategy's Business Intelligence platform, DataTern will not be able to prove infringement because of the NY Markman Order's construction of the phrase "to create at least one interface object." (SOF ¶ 49.) DataTern's draft thereby sets forth a proposed judgment of non-infringement based on a single claim limitation construed in the New York action:

Applying the NY Markman Order, MicroStrategy's Business Intelligence platform does not infringe the '502 Patent as it does not meet the "to create an interface object" limitation in that it does not "generate code for at least one class and instantiate an object from that class".

(SOF ¶ 50.) DataTern's proposed judgment limits its concession of noninfringement to the term "to create an interface object." (SOF ¶ 51.)

This motion, seeking judgment of noninfringement on multiple claim limitation, and further seeking fees for DataTern's vexatious litigation conduct, followed.

#### **IV. TECHNICAL BACKGROUND**

##### **A. Technology Tutorial<sup>4</sup>**

###### **1. Object-Oriented Programming**

Object-oriented programming is a programming paradigm that uses software "objects." Specifically, object-oriented programming organizes a software application into a collection of objects, which are defined by "classes." Each class defines what types of data

---

<sup>4</sup> This section is intended to present the underlying technology in a non-controversial manner to assist the Court.

an object will store (“attributes”) and what tasks (“methods”) the object can perform, if any.

For instance, in an email application, an example of a class is a Message class. Another example could be a Contact class. Attributes of a message class could be, *e.g.*, ‘Message ID,’ ‘Subject,’ and/or ‘Sender.’ The Message class also defines the data types of each attribute. For instance, a ‘Message ID’ attribute might be stored as a number; whereas the ‘Subject’ might be stored as a string of characters. And, at times, an attribute of a class might be stored using a type corresponding to another class. For instance, in the above example, the Message class attribute ‘Sender’ might be stored using the Contact class, which has its own attributes and methods. An example task (or method) for the Message class could be a ‘Send’ method, which would define the necessary operations to send a particular message.

<u>Message Class</u>		<u>Contact Class</u>	
Attributes	Type	Attributes	Type
Message ID	Number	First Name	String of Characters
Subject	String of Characters	Last Name	String of Characters
Sender	Contact class →	Phone	Number
<b>Methods</b>		<b>Methods</b>	
Message.Send ()		Contact.Delete()	

“Objects” are specific instances of a class that hold actual values and can carry out functionality defined in the class methods. For instance, during execution, an email application might instantiate a new Message object, which would conform to the template defined by the Message class. The instantiated Message object can hold actual values corresponding to each attribute, *e.g.*, ‘101,’ ‘Re: Thursday’s Meeting,’ and ‘Bill Jones’ (for attributes Message ID, Subject, and Sender). While the email application is running, it may instantiate any number of Message objects, all of which must conform to the template defined by the Message class. Two example Message objects appear below.

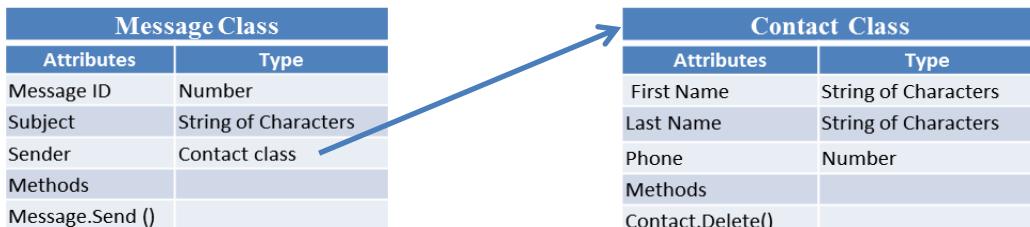
**Message Object #1**

Attributes	Value
Message ID	101
Subject	Meeting at 10:00 a.m.
Sender	Robert Jones
<b>Methods</b>	
Message.Send ()	

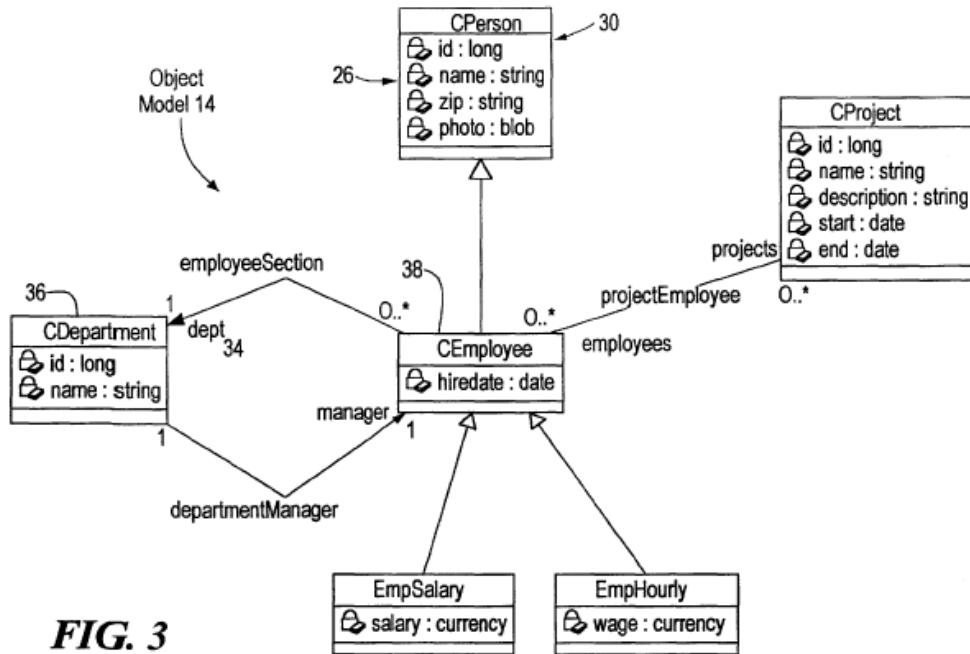
**Message Object #2**

Attributes	Value
Message ID	102
Subject	Meeting Cancelled
Sender	Bill Jones
<b>Methods</b>	
Message.Send ()	

Since an object-oriented application consists of any number of objects, all defined by classes, an “object model” can be used to define all classes that make up an object-oriented application and the relationships between those classes. For instance, in the above example, one such class relationship is that between the ‘Sender’ attribute of the Message class and the Contact class. Namely, a relationship exists between these classes because each instantiated Message object will have a Sender value that points to an instantiated Contact object. That relationship appears in the below illustration, which depicts a small portion of an object model for an email application.



The '502 patent Fig. 3 illustrates an exemplary object model for an application that maintains information about a company's employees.



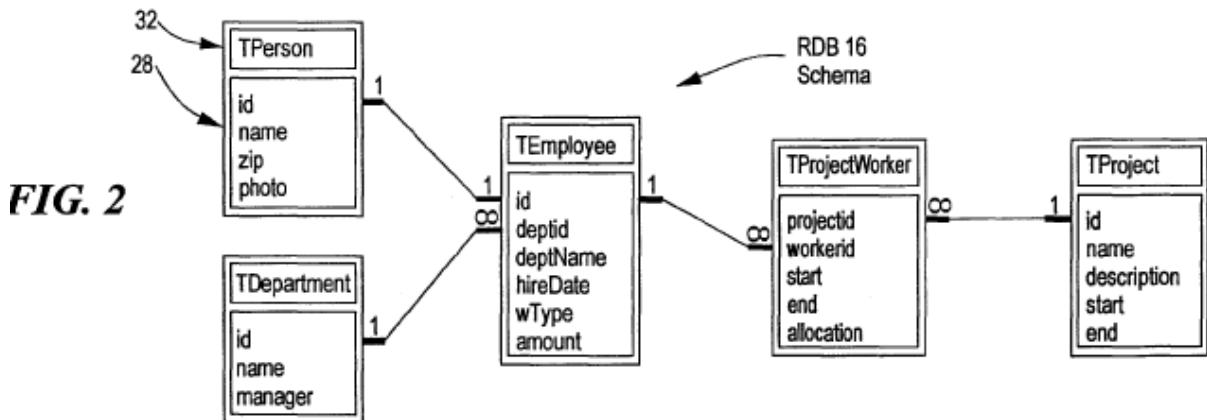
## 2. Relational Databases

A database is a collection of stored information. Data in a database is organized into “tables.” Each table has rows and columns: the rows specify particular data items (e.g., a particular email message), and the columns specify information about those items (sender, recipient, subject, etc.). Databases, including relational databases, are used to store data required by an object-oriented software application.

Some databases are relational, meaning that multiple tables in the database are related to one another. For example, some email programs keep track of contact information. Emails and contacts are stored in separate tables. While the columns in the email table might include sender, recipient, and subject, the columns in the contacts table might include name, address, phone, and email address. The “sender” column in the email table may relate to the “email address” column in the contacts table. In a relational database, a user can look up a sender’s contact information by identifying the particular sender individual in the contacts database.

A database can be described in terms of its structure, which is also called a schema. A database schema defines individual tables in the database, the data attributes in each table,

and relationships among tables. The '502 patent's Fig. 2 provides an exemplary relational database schema:



As represented by the schema in the example, the database includes, *e.g.*, a "TEmployee" table and a "TPerson" table, which represent employees and people, respectively. Since all employees are also people, the tables are related; the common attribute "id" allows one to look up the "TPerson" that corresponds to a particular "TEmployee."

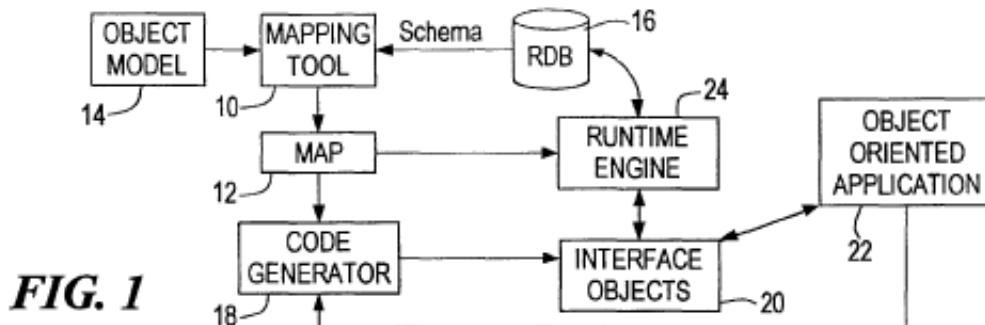
## B. The Disclosure of the '502 Patent

The '502 patent is directed to "interfacing object oriented software applications with relational databases." (SOF ¶ 52.) The '502 patent acknowledges that the "need for interfacing object oriented software applications with relational databases is well known" and multiple prior art methods existed for doing so. (SOF ¶ 53.) Thus, the patent discloses a specific alleged solution to this perceived need which involves "mapping between an object model and a relational database." (SOF ¶ 54.)

At a high level, the mapping described in the patent merely means that class attributes are associated with analogous table fields. For example, the patent's Fig. 3 shows a CPerson class with a "name" attribute. The patent's Fig. 2 shows a TPerson table with a "name" field. A mapping, as described in the patent, would indicate that there is an association between

CPerson.name and TPerson.name. This allows the object-oriented application to construct a set of objects based on information stored in a database.

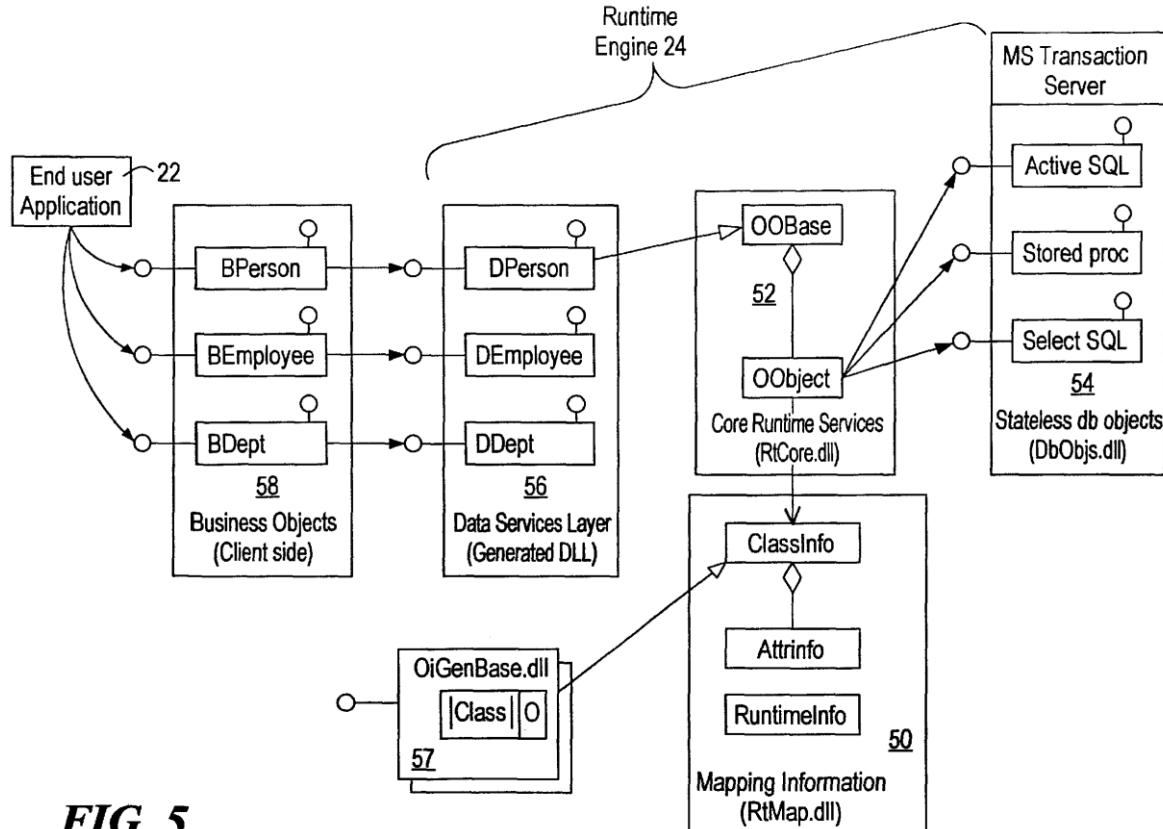
In the words of the patent, a “database schema, object model, and mapping are employed to provide interface objects that are utilized by the object oriented software application to access the relational database.” (SOF ¶ 55.) The disclosed technology was intended to “provide[] transparent access to the relational database.” (SOF ¶ 56.) That way, the software application, without detailed knowledge of the underlying database, could still programmatically store and retrieve information from the database for use in the application. (SOF ¶ 57.) It is the map, rather than the software application, that has the detailed information about the underlying database.



The patent uses Figure 1 (above) to illustrate the use of a map to generate interface objects that are employed by a runtime engine and an object oriented software application to access a relational database. (SOF ¶ 58.) During a software development phase, a mapping tool is used to create a map between (a) an object model corresponding to a software application and (b) the underlying relational database, which contains information relevant to the application. (SOF ¶ 59.) Specifically, the mapping consists of, at least, mapping attributes within software classes associated with the software application to corresponding database table columns. (SOF ¶ 60.)

The resulting map is then used by a code generator to generate new software code for an “interface object.” (SOF ¶ 61.) Software code defining at least one new classes is created

by examining the relationships defined in the map. (SOF ¶ 62.) These generated classes correspond to classes within the object oriented application, as shown in the patent's Fig. 5 (SOF ¶ 63.):



**FIG. 5**

As shown above, each of the interface objects 56 corresponds to a single class 58 in the object-oriented software application. During execution of the software application, the application calls the interface objects to access the relational database via a runtime engine. (SOF ¶ 64.)

### C. Claim 1 of the '502 Patent

The '502 patent includes two independent claims, 1 and 11. Exemplary claim 1 reads as follows:

A method for interfacing an object oriented software application with a relational database, comprising the steps of:

- (a) selecting an object model;
- (b) generating a map of at least some relationships between schema in the database and the selected object model;

(c) employing the map to create at least one interface object associated with an object corresponding to a class associated with the object oriented software application; and

(d) utilizing a runtime engine which invokes said at least one interface object with the object oriented application to access data from the relational database.

The preamble introduces the purpose of the claim: to allow an object-oriented software application to interface with a relational database.

The claim's first step requires "selecting an object model." The object model describes the classes used in that software application, as well as the relationships between the classes. (SOF ¶ 64.) The '502 specification also explains that "the object model 14 is a template that has a predetermined standardized structure." (SOF ¶ 65.) A predetermined structure, or a standardized way for an object model to represent classes, their attributes, and relationships, facilitates the automatic creation of a map in the next step.

Claim 1 next requires generating a map of relationships between the database schema (which describes the structure of the database) and the object model (which describes the classes in the object-oriented software application). This map associates attributes within classes in the software application to corresponding database table columns. (SOF ¶ 66.)

After the map is generated, the claim requires that the map be used to create (i.e., generate code for) an "interface object." (SOF ¶ 67.) The interface object will later be used "to access data from the relational database." The interface object must be "associated with an object corresponding to a class associated with the object oriented software application." This means, as shown in the patent's Fig. 5, that for each interface object, there is a single corresponding class in the object-oriented software application.

The final limitation of claim 1 requires a runtime engine to invoke an interface object "with the object oriented program" to access data from the database. The interface object, according to the claim, is ultimately responsible for reading and writing data to the database. (SOF ¶ 68.)

#### D. The New York Claim Constructions and Summary Judgment Order

In DataTern's New York action, the court construed four disputed terms, all of which appear in both of the '502 patent's independent claims. (SOF ¶ 69.) Those terms and constructions are as follows:

- “**object model**” – “A template with a predetermined standardized structure both relating to an object-oriented software application and including object classes and inheritance relationships among classes.”
- “**to create at least one interface object**” - To generate code for at least one class and instantiate an object from that class, where the object is not part of or generated by the object oriented application and is used to access the database.”
- “[at least one interface object] **associated with an object corresponding to a class associated with the object oriented software application**” - “The interface object corresponds to an object instantiated from a single class associated with the object oriented software.”
- “**runtime engine**” – “Software that (i) the object oriented software application depends on to run, (ii) must be running to execute the object oriented software application (iii) uses the map in its processing, and (iv) is not part of the object oriented software application.”

(SOF ¶ 70(A).)

The New York court relied on its construction of three of these four terms in granting SAP's motion for summary judgment of non-infringement. (SOF ¶ 71.) Specifically, SAP moved for summary judgment alleging DataTern could not prove SAP's accused products satisfied any of the above constructions. (SOF ¶ 72.) DataTern conceded that, applying the court's claim constructions, DataTern could not prove SAP's products satisfied the “object model,” “to create at least one interface object,” or “runtime engine” limitations. (SOF ¶ 73.) Thus, SAP requested the court grant summary judgment on at least four bases, each of which was based on one of the three terms DataTern conceded SAP's products could not meet (SOF ¶ 74) (emphasis added):

1. [SAP Accused Product] does not permit selecting an “**object model**” because the accused “**object model**” does not contain inheritance relationships among classes.

2. [SAP Accused Product] also does not permit selecting an “**object model**” because the alleged “classes” in the accused object model do not include behaviors.
3. [SAP Accused Product] does not “**create at least one interface object**” because it “d[oes] not generate code for at least one class and then instantiate a query object from that class.”
4. [SAP Accused Product] does not have a “**runtime engine**,” at least because its alleged “runtime engine” is not one that the object oriented software application depends on to run.

The New York court granted a judgment of no infringement in SAP’s favor on each of the above grounds, stating “Plaintiffs’ motions for summary judgment of noninfringement are GRANTED on the basis of defendant’s concessions that there are four separate reasons why it cannot prove infringement in light of this Court’s Markman decision of August 24, 2012.” (SOF ¶ 75.)

#### **E. DataTern’s Infringement Contentions**

As explained above (§ III.A), DataTern has served two sets of infringement contentions in this case. After DataTern served its initial set, MicroStrategy moved to compel more specific contentions, pointing out that DataTern’s contentions did not adequately place MicroStrategy on notice of DataTern’s theories. (SOF ¶ 76.) The Court agreed, and ordered DataTern to serve supplemental contentions that specifically identify “where each asserted claim element is found within each accused product or combination *and supporting evidence*.” (SOF ¶ 25 (emphasis added).) DataTern subsequently served amended contentions (SOF ¶ 26), which failed to cure many of the deficiencies found in DataTern’s original contentions. DataTern’s amended contentions are summarized below.

##### **1. “object model”**

DataTern’s amended infringement contentions allege that a so-called “Logical Data Model” referred to in MicroStrategy’s product literature constitutes the claimed “object model.” (SOF ¶ 77.) As the evidence cited in DataTern’s contentions demonstrates,

MicroStrategy's Logical Data Model is a user's description of how various components of that user's business data relate to one another. (SOF ¶ 78.) DataTern does not allege the Logical Data Model can consist of software classes or that it pertains to an object oriented software application, as required by the New York court's claim constructions. (SOF ¶ 79.) Further, DataTern's infringement contentions do not allege that MicroStrategy's Logical Data Model constitutes "a template with a predetermined standardized structure," as expressly required by the patent and by the New York court's claim constructions. (SOF ¶ 80.)

**2. "create at least one an interface object"**

DataTern's infringement contentions assert that MicroStrategy products "create at least one interface object" because:

as the cited evidence indicates, when such a client application requires access to data from a relational database, the MicroStrategy Intelligence Server component of the MicroStrategy ***Business Intelligence platform instantiates an interface object*** employing the persisted metadata repository.

(SOF ¶ 81) (emphasis added). In other words, DataTern's allegations contend that MicroStrategy's products "create . . . an interface object" because the products instantiate a software object from an existing software class. DataTern does not contend that MicroStrategy's products can generate software code constituting an "interface object." (SOF ¶ 82.) Also, DataTern did not provide any "cited evidence" to support this theory or specifically to identify which alleged instantiated object within MicroStrategy's software products constitutes the alleged "interface object." (SOF ¶ 83.)

**3. "associated with an object corresponding to a class associated with the object oriented software application"**

DataTern also alleges MicroStrategy's products create an interface object that is "associated with an object corresponding to a class associated with the object oriented software application" because:

As the cited evidence indicates, ***components in the "Develop" layer*** of the MicroStrategy Business Intelligence platform—including MicroStrategy

Desktop Analyst & Designer, MicroStrategy Architect, and MicroStrategy SDK—*ensure that client applications created by MicroStrategy Incorporated’s licensees, customers, and/or users* and designed to access data from a relational database by interacting with the MicroStrategy Business Intelligence platform *will also contain similar sets of associated objects* that are instances of classes defined in the client applications. See Single Unified Architecture figure above.

(SOF ¶ 84.) But nowhere do DataTern’s contentions disclose how the components in the “Develop” layer “ensure that client applications created by MicroStrategy Incorporated’s licensees, customers, and/or users . . . will also contain similar sets of associated objects.” The contentions disclose no information how any such components work, nor do they disclose which “client applications” allegedly contain “similar sets of associated objects,” nor do they disclose how the sets of associated objects in the client applications are allegedly “similar,” nor do they disclose any interaction between components in the “Develop” layer and “client applications” that would cause the client applications to include “similar sets of associated objects.”

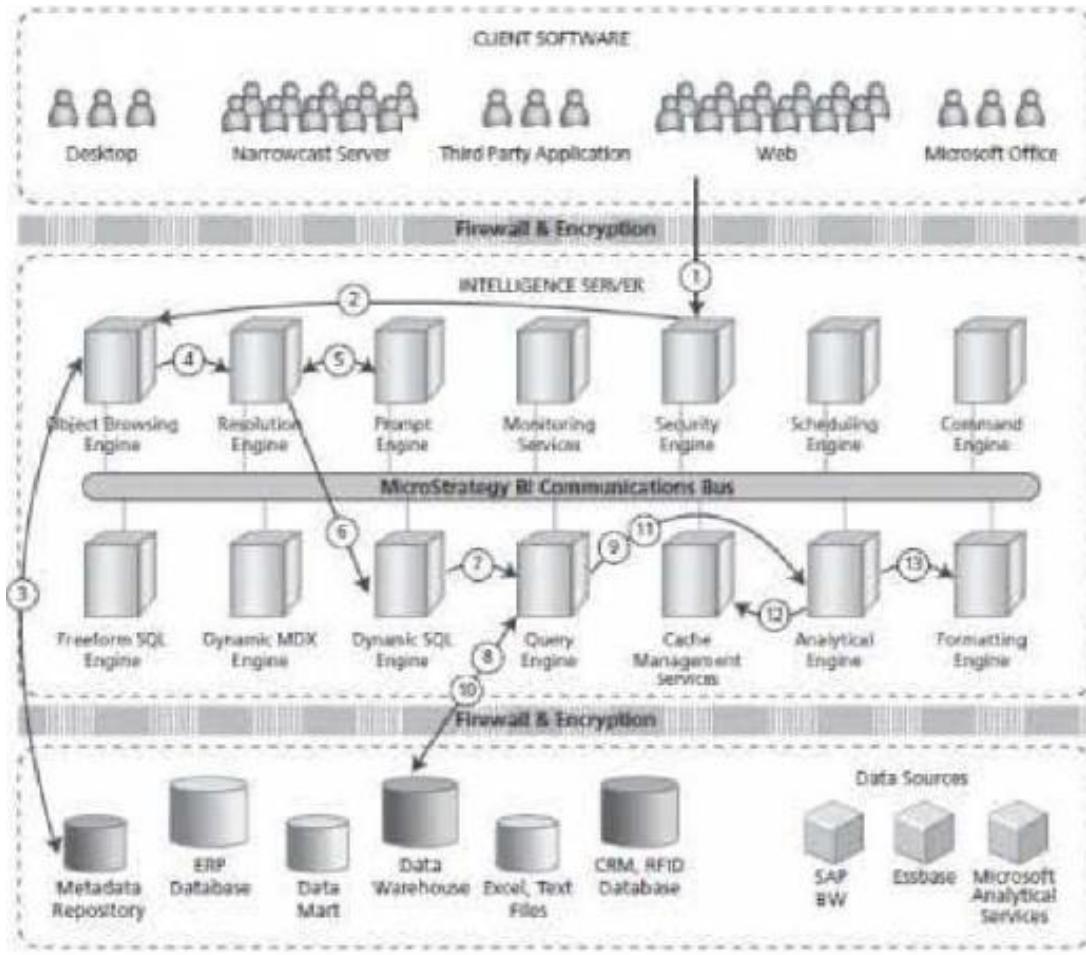
#### **4. “runtime engine”**

DataTern’s infringement contentions allege that MicroStrategy’s Business Intelligence Server constitutes the claimed “runtime engine.” (SOF ¶ 85.) DataTern does not assert that any object-oriented software application depends on the runtime engine to run.

#### **5. “interface object . . . to access data from the relational database**

Finally, DataTern’s infringement allegations claim MicroStrategy Products use an “interface object . . . to access data from the relational database.” (SOF ¶ 86.) DataTern’s amended contentions allege MicroStrategy satisfies this limitation because “the MicroStrategy Intelligence Server component of the MicroStrategy Business Intelligence platform invokes the required Metadata Objects (interface objects) to access data from the relational database.” (SOF ¶ 87.) DataTern supports its assertion with one high-level MicroStrategy diagram. (SOF ¶ 88.) As shown below, the diagram depicts a Query Engine,

not a Metadata Object, as the component responsible for accessing the alleged “relational database” (the “Data Warehouse”).



(SOF ¶ 89.) The diagram’s caption also states that “the [MicroStrategy] QueryEngine connects to the relevant data source and runs the query.” (SOF ¶ 90.)

## V. THE COURT SHOULD GRANT SUMMARY JUDGMENT OF NONINFRINGEMENT

### A. Legal Standards

#### 1. Summary Judgment

Summary judgment under Rule 56(a) is appropriate where “the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to a judgment as a matter of law.” Fed. R. Civ. P. 56(a). Because DataTern has the ultimate burden to prove infringement, MicroStrategy can carry its burden on summary judgment by demonstrating the

absence of evidence to support DataTern’s infringement allegations. *Regents of Univ. of Cal. v. Dako N. Am., Inc.*, No. C 05-03955, 2009 WL 1083446, at \*5 (N.D. Cal. Apr. 22, 2009) (internal quotations omitted) (“On an issue for which the opposing party will have the burden of proof at trial, the moving party need only point out that there is an absence of evidence to support the nonmoving party’s case.”). Although a nonmovant may oppose a summary judgment motion by stating that “it cannot present facts essential to justify its opposition,” Fed. R. Civ. P. 56(d), a submission under this rule must “demonstrate that movant has been diligent in conducting discovery, and show ‘good cause’ why the additional discovery was not previously practicable with reasonable diligence.” *Simas v. First Citizens' Fed. Credit Union*, 170 F.3d 37, 45 n.4 (1st Cir. 1999).

## **2. Collateral Estoppel Effect of Claim Construction Rulings**

“[W]here a determination of the scope of patent claims was made in a prior case, and the determination was essential to the judgment there on the issue of infringement, there is collateral estoppel in a later case on the scope of such claims, i.e., the determined scope cannot be changed.” *Molinaro v. Fannon/Courier Corp.*, 745 F.2d 651, 655 (Fed. Cir. 1984)); *see also Pfaff v. Wells Electronics, Inc.*, 5 F.3d 514, 518 (Fed. Cir. 1993) (“The prior claim interpretation has issue preclusive effect in the present case insofar as it was necessary to the judgment of noninfringement in the previous case.”); *Biovail Laboratories Intern. v. Intelgenx Corp.*, Civ. No. 09-605-LPS, 2010 WL 5625746 at \*4 (D. Del. December 27, 2010) (“More specifically, the Federal Circuit has made clear that collateral estoppel applies to a claim construction ruling upon the entry of a final judgment in which the court construes the claim at issue, as long as the claim construction was ‘essential’ to the final judgment.”).

**B. The Court’s Should Grant Summary Judgment of Noninfringement Because DataTern Concedes that the Accused Products Do Not Create an Interface Object**

There is no dispute that MicroStrategy’s products do not satisfy the claim requirement of “creat[ing] an interface object” under the New York court’s claim constructions. Nor is there a dispute that judgment should enter on this ground. (See Ex. A.) DataTern’s proposed judgment provided to MicroStrategy provides in part as follows:

18. Accordingly, in order to simplify these proceedings, DataTern, Inc. (“DataTern”) hereby offers that if the *NY Markman Order is applied to MicroStrategy’s Business Intelligence platform, DataTern will not be able to prove infringement because of the NY Markman Order’s construction of the phrase “to create at least one interface object”*.

19. The phrase “to create at least one interface object” appears within subpart (c) of claim 1 and within subpart (b) of claim 10 of the ‘502 Patent. Claim 1 of the ‘502 Patent requires “employing the map to create at least one interface object associated with an object corresponding to a class associated with the object oriented software application.” Claim 10 includes similar language. The NY Markman Order construed the phrase “to create at least one interface object” as meaning “to generate code for at least one class and instantiate an object from that class, where the object is not part of or generated by the object oriented application and is used to access the database”. *DataTern offers to concede that Microstrategy’s Business Intelligence platform does not meet the “to create at least one interface object” limitation because the NY Markman Order construed that limitation to require “generatfion off code for at least one class” from which the interface object is instantiated.*

(*Id.* at 8 (emphasis added).) MicroStrategy agrees that DataTern has not shown that the accused products “generate code for at least one class,” and therefore, MicroStrategy is entitled to judgment of noninfringement in its favor on this ground.

Moreover, DataTern’s infringement contentions do not allege that this limitation, as construed by the New York court, is satisfied. Specifically, nothing in DataTern’s infringement contentions suggests that MicroStrategy products generate code for a class from which an interface object is instantiated. (See D.I. 72-1 at 7 (arguing, in relevant part, that only that the “MicroStrategy Intelligence Server . . . instantiates an interface object employing the persisted metadata repository”; SOF ¶ 70(A) (setting out the New York court’s

construction requiring that “creat[ing] an interface object” involves generation of code).)

Apart from DataTern’s concession, then, DataTern has not satisfied its burden of providing evidence to substantiate its infringement contentions. (See July 2, 2012 Order.)

**C. The Court Should Grant Summary Judgment on Two Further Grounds Mandated By the New York Court’s Claim Constructions**

**1. DataTern Is Bound By Those of the New York Court’s Claim Constructions That Were Essential to Its Judgment**

Beyond DataTern’s concession, two additional grounds for noninfringement arise directly out of the New York court’s final judgment. This Court should grant summary judgment of noninfringement on these grounds as well, to preserve judicial resources in the event the Federal Circuit affirms the New York court’s claim construction on one of these additional grounds.

Collateral estoppel precludes DataTern from contesting the construction of three claim limitations that were essential to the final judgment in New York: “object model,” “create at least one interface object,”<sup>5</sup> and “runtime engine.” *See e.g., Molinaro*, 745 F.2d at 655. SAP moved for summary judgment of no infringement based on multiple grounds, each explicitly depending on the construction of one of these terms. (*See supra*, § IV.D.) DataTern conceded the SAP accused products could not satisfy these limitations, as construed. *Id.* The New York court—granting SAP’s motion for summary judgment—based its judgment explicitly on the above-enumerated grounds, stating “Plaintiffs’ motions for summary judgment of noninfringement are GRANTED on the basis of defendant’s concessions that there are ***four separate reasons why it cannot prove infringement in light of this Court’s Markman decision*** of August 24, 2012.” *Id.* (emphasis added). Because the New York court’s judgment of no infringement unambiguously relied on three claim constructions issued in that case, those three constructions have a preclusive

---

<sup>5</sup> DataTern has conceded noninfringement based on this term.

effect here. *See Pfaff*, 5 F.3d at 518 (“The prior claim interpretation has issue preclusive effect in the present case insofar as it was necessary to the judgment of noninfringement in the previous case.”);

## **2. The Accused Products Do Not Satisfy the “Object Model” Limitation**

All asserted claims of the ’502 patent require selection of an “object model.” DataTern cannot demonstrate MicroStrategy satisfies this claim limitation under the New York construction. The New York court construed “object model” to mean “a template with a predetermined standardized structure both relating to an object-oriented software application and including object classes and inheritance relationships among classes.” (*See supra*, § IV.D.) DataTern conceded that the SAP accused product did not include an object model that contains “inheritance relationships among classes,” and also that the alleged “classes” do not “include behaviors.” (*Id.*)

Even more so, the accused MicroStrategy products do not satisfy the “object model” limitation. The alleged “object model”—MicroStrategy’s Logical Data Model—is merely a user’s designation of the relationship between different types of business data. (D.I. 72-1 at 5 (“A logical data model is a logical arrangement of data as experienced by the general user or business analyst.”) It does not “relat[e] to an object-oriented software application,” nor does it contain “inheritance relationships among classes.” Nor does DataTern explain what it contends the “classes” in MicroStrategy’s software to be, or whether or not such alleged classes include “behaviors.” (*See generally* D.I. 72-1.)

Because the alleged object model has nothing to do with an object-oriented software application, but is rather a user’s own depiction of relationships between different types of business data, the Court should grant summary judgment of no infringement based on DataTern’s failure to show the selection of an “object model,” as required by all claims of the ’502 patent.

**3. The Accused Products Do Not Satisfy the “Runtime Engine” Limitation**

The claims of the '502 patent also require a “runtime engine.” The New York court construed this phrase to mean “software that (i) the object oriented software application depends on to run, (ii) must be running to execute the object oriented software application, (iii) uses the map in its processing, and (iv) is not part of the object oriented software application.” (*See supra*, § IV.D.) The New York court granted summary judgment in favor of SAP because SAP’s product “does not have a ‘runtime engine,’ at least because its alleged ‘runtime engine’ is not one that the object oriented software application depends on to run.” (*See supra*, § IV.D.)

Nowhere does DataTern even allege that the accused runtime engine is software that the object oriented software application depends on to run.” Nothing in DataTern’s infringement contentions suggests that MicroStrategy’s alleged “runtime engine” (the MicroStrategy Intelligence Server) must be running to execute a user’s application. Because DataTern has failed to provide any evidence whatsoever regarding how MicroStrategy’s products satisfy these claim requirements, a judgment that MicroStrategy cannot satisfy this claim limitation under the New York construction is also proper. *Nomadix, Inc. v. Hewlett-Packard Co.*, 2012 WL 682874, \*2 (C.D. Cal. Mar. 1, 2012) (granting motion for summary judgment where alleged infringer “pointed to an absence of [patentee’s] evidence in support of the relevant claim limitations”).

**D. The Court Should Further Grant Summary Judgment on Three Additional Grounds, Because DataTern’s Infringement Contentions Do Not Establish Infringement**

**1. Summary Judgment Is Appropriate in Light of DataTern’s Infringement Contentions**

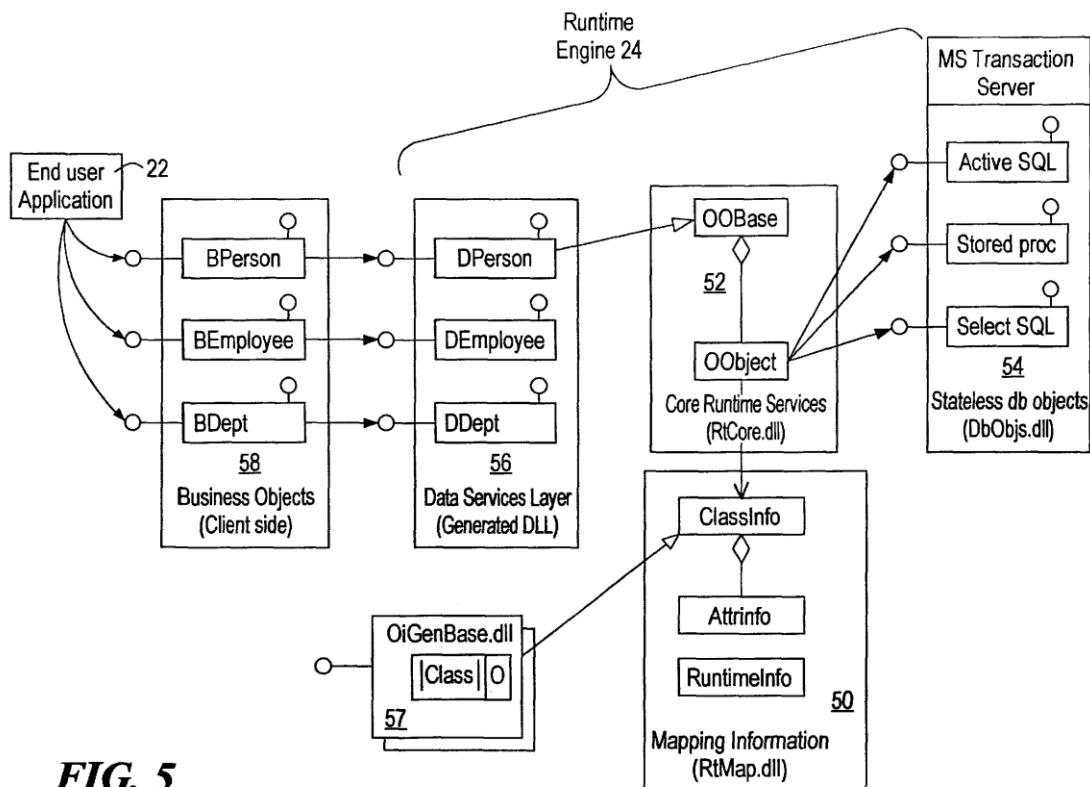
The Court should rely on DataTern’s infringement contentions when determining whether DataTern can satisfy its summary judgment burden. Under this Court’s Order,

DataTern was required to serve amended infringement contentions that set forth DataTern’s “good faith basis (under Rule 11) that each accused product (or combination of products) meets each limitation of an asserted claim,” including “the theory under which it is alleged to be liable.” (July 2, 2012 Electronic Order.) “[D]efendants are entitled to know . . . how each asserted claim limitation is allegedly met by each accused product.” (*Id.*) Thus, if DataTern’s infringement contentions do not set forth a *prima facie* case of infringement, the Court should grant summary judgment. *See Nomadix*, 2012 WL 682874, at \*2-3 (granting summary judgment based on legally insufficient infringement contentions); *see also O2 Micro Int'l Ltd. v. Monolithic Power Sys. Inc.*, 467 F.3d 1355, 1366 (Fed. Cir. 2005) (affirming a district court’s grant of summary judgment based on the patentee’s failure to serve adequate infringement contentions). Because, as explained below, DataTern’s infringement contentions (even if true) do not establish that several claim limitations are satisfied, the Court should grant judgment of noninfringement on these additional grounds.

DataTern should not be heard to argue that it needs further discovery. As explained above, MicroStrategy’s source code has been available for inspection since August 10, 2012. (*See supra* §III.A.) During the October 5, 2012 hearing in this matter, the Court stated, “[I]f you think you need to [inspect MicroStrategy’s source code], the time to do it is now.” (*Id.*) DataTern did not do so, and its lack of diligence precludes it from opposing summary judgment based on an alleged need for discovery now. *Simas v. First Citizens' Fed. Credit Union*, 170 F.3d 37, 45 n.4 (1st Cir. 1999) (holding that a litigant wishing to rely on Rule 56(d) (then Rule 56(f)) must show that it “has been diligent in conducting discovery, and show ‘good cause’ why the additional discovery was not previously practicable with reasonable diligence”).

**2. DataTern Cannot Show That the Accused Products Create an Interface Object “Associated With An Object Corresponding To A Class Associated With The Object Oriented Software Application”**

The independent claims of the '502 patent require that the claimed interface object be “associated with an object corresponding to a class associated with the object oriented software application.” In other words, the interface object (which is alleged to be present in MicroStrategy’s products) must be associated with *other* objects found in a different software application. The patent’s Figure 5 depicts this arrangement:



**FIG. 5**

As shown above, in the patent, the “generated DLL” objects (interface objects) 56 are each associated with “client side” objects (objects in an object-oriented software application) 58. But DataTern has provided no evidence that there is any such association between objects in MicroStrategy’s software and objects in an object-oriented software application, as required by the claims.

The Court’s July 2, 2012 Order required DataTern to provide “supporting evidence” in connection with its allegations. (July 2, 2012 Electronic Order.) With respect to this claim limitation, DataTern’s contentions state:

[A]s the cited evidence indicates, components in the ‘Develop’ layer of the MicroStrategy Business Intelligence platform. . . ensure that client applications created by MicroStrategy Incorporated’s licensees, customers, and/or users . . . will also contain similar sets of associated objects.

(D.I. 72-1 at 8.) But in reality, DataTern provided no “cited evidence” to support its allegation. DataTern did not, in particular, provide any evidence that the components in the “Develop” layer allegedly “ensure that client applications created by MicroStrategy Incorporated’s licensees, customers, and/or users . . . will also contain similar sets of associated objects.” DataTern’s contentions do not analyze any “client application” to determine whether they do, in fact, contain “similar sets of associated objects.” Because DataTern has supplied no evidence that the accused products create an interface object “associated with an object corresponding to a class associated with the object oriented software application,” the Court should grant summary judgment on this ground as well. *See Nomadix*, 2012 WL 682874, at \*2 (“[C]onclusory allegations are insufficient to create genuine issues of material fact as to the claim limitations, and [the accused infringer] has therefore met its initial summary judgment burden”); *see also Moore U.S.A., Inc. v. Standard Register Co.*, 229 F.3d 1091, 1112 (Fed. Cir. 2000) (affirming lower court’s summary judgment where patentee’s “allegations” were “entirely lacking in factual support”).

### **3. DataTern Cannot Show that MicroStrategy’s Products Include a “Code Generator”**

Claim 10 of the ’502 patent requires a “code generator” to create an interface object. It is now undisputed that the accused products do not generate code to create interface object. (See Ex. A at 12, ¶ 1 (“MicroStrategy’s Business Intelligence platform . . . does not ‘generate code for at least one class and instantiate an object from that class.’”)) For this independent

reason, the Court should grant summary judgment of non-infringement of claim 10, as well as the claims that depend from claim 10. *See Monsanto Co. v. Syngenta Seeds, Inc.*, 503 F.3d 1352, 1359 (Fed. Cir. 2007) (“One who does not infringe an independent claim cannot infringe a claim dependent on (and thus containing all the limitations of) that claim.”).

**4. DataTern Cannot Show that MicroStrategy’s Products Invoke an Interface Object “With the Object Oriented Application”**

Claim 1 of the ’502 patent requires a runtime engine that “invokes said at least one interface object *with the object oriented application.*” (’502 patent at 8:1-2 (emphasis added).)<sup>6</sup> Nowhere do DataTern’s infringement contentions explain how an interface object is allegedly invoked “with the object oriented application.” To the contrary, as noted above, DataTern argues that “the MicroStrategy Intelligence Server component of the MicroStrategy Business Intelligence platform [the alleged runtime engine] invokes the required Metadata Objects (interface objects) to access data from the relational database.” (D.I. 72-1 at 11). The alleged “object oriented application” apparently plays no role, according to DataTern’s infringement theory. Because DataTern’s infringement theory ignores the express language of claim 1 requiring that the interface object be invoked “with the object oriented application,” the Court should grant summary judgment at least of no infringement of claim 1 of the ’502 patent, as well as claims 2-9, which depend from claim 1. *See Monsanto*, 503 F.3d at 1359 (Fed. Cir. 2007).

**VI. THE COURT SHOULD DECLARE THIS TO BE AN EXCEPTIONAL CASE AND AWARD MICROSTRATEGY ITS REASONABLE ATTORNEYS’ FEES**

**A. Legal Standards**

Pursuant to 35 U.S.C. § 285, the Court may award reasonable attorneys’ fees to a prevailing party in “exceptional cases.” Litigation that is objectively baseless and brought in bad faith justifies an exceptional case finding and award of attorneys’ fees. *MarcTec, LLC v.*

---

<sup>6</sup> The other independent claim of the ’502 patent, claim 10, does not include this requirement.

*Johnson & Johnson*, 664 F. 3d 907, 916 (Fed. Cir. 2012). “Under this standard, a patentee’s case must have no objective foundation, and the plaintiff must actually know this.” *Id.* (quotation marks and citation omitted). Infringement allegations are objectively baseless when “no reasonable litigant could reasonably expect success on the merits.” *Highmark, Inc. v. Allcare Health Mgmt. Sys.*, 687 F.3d 1300, 1309 (Fed. Cir. 2012) (internal quotation marks omitted). Here, DataTern could not reasonably expect success on the merits; indeed, it served infringement contentions that did not establish infringement even under DataTern’s own claim construction proposals. The Court should therefore grant MicroStrategy its reasonable attorneys’ fees.

**B. The Court Should Declare This to Be an Exceptional Case and Award MicroStrategy Its Reasonable Attorneys’ Fees for Defending Against DataTern’s Baseless Allegations**

As explained below, DataTern’s assertion of the ’502 patent against MicroStrategy was objectively baseless, “and the plaintiff . . . actually kn[e]w this.” *See MarcTec*, 664 F.3d at 916. Thus, the Court should declare this to be an exceptional case and award MicroStrategy its reasonable attorneys’ fees related to MicroStrategy’s defense of this action. A “fair estimate” of those fees are set forth in the accompanying declaration of Benjamin K. Thompson (hereinafter “Thompson Decl.”). *See* Fed. R. Civ. P. 54(d)(2)(B)(iii).

**1. DataTern’s Allegations That Microstrategy’s Products Satisfied The “Object Model” Limitation Are Baseless**

DataTern must reasonably have known that an “object model” must include software classes. An “object” inherently involves the notion of classes. The New York court agreed that “***there is no way*** to read in a possibility that an object model is not connected to or representing classes; it is mandatory, not discretionary.” (SOF ¶ 70(B) (emphasis added).) The Court noted that the claimed mapping between an object model and schema in a relational database “is central to the fundamental purpose of the patent.” (*Id.*) The Court stated that the “specifications make clear that this process contemplates the mapping of ***class attributes***

to schema in relational databases.” (*Id.*; *see also* ’502 patent at 2:43-48.) “Therefore,” the court concluded, “to fulfill the central purpose of the patent, ***the object model must include classes.***” (*Id.* (emphasis added).) DataTern could not reasonably have believed that an “object model” would not require the inclusion of classes.

There can be no genuine dispute that the Logical Data Model—a user’s description of relationships between business data—has nothing to do with classes. DataTern knew this, and its infringement contentions are bereft of any claim that a Logical Data Model includes classes. Because DataTern’s infringement theory regarding “object model” was objectively baseless, and DataTern must have known that, the Court should declare this to be an exceptional case and grant MicroStrategy its reasonable attorneys’ fees. *See MarcTec, LLC v. Johnson & Johnson*, 664 F. 3d 907, 920 (Fed. Cir. 2012) (upholding district court’s grant of attorney’s fees where Plaintiff “ignored language in the specification and statements made during prosecution that directly contradicted the plain meaning arguments it advanced.”).

**2. DataTern Lacks Any Basis for its Allegations that the Accused Products Create an Interface Object “Associated With An Object Corresponding To A Class Associated With The Object Oriented Software Application”**

As explained above, each of the asserted claims requires the creation of an interface object “associated with an object corresponding to a class associated with the object oriented software application.” (*See supra*, § V.D.2.) DataTern’s infringement theory requires that there be an object in MicroStrategy’s product that corresponds to objects in a separate object-oriented software application. (*See* D.I. 72-1 at 7-8 (“[C]omponents in the ‘Develop’ layer of the MicroStrategy Business Intelligence platform . . . ensure that client applications . . . will also contain similar sets of associated objects.”) But DataTern has never explained why it believes there to be such an association. (*See supra*, § V.D.2.) Because DataTern has not provided a “good faith basis (under Rule 11) that each accused product (or combination of products) meets each limitation of an asserted claim” as required by this Court’s July 2, 2012

Order, the Court may presume that DataTern had no such basis, and should declare this to be an exceptional case.

**3. DataTern’s Allegations that the Accused Products Invoke an Interface Object “With the Object Oriented Application” Are Baseless**

Claim 1 of the ’502 patent requires invoking “at least one interface object *with the object oriented application.*” As previously explained (*see supra* § V.D.4), DataTern has never articulated a basis for alleging that this claim limitation is satisfied. DataTern’s infringement contentions state that the alleged runtime engine invokes the alleged interface object directly—without involving the claimed “object oriented application.” Because DataTern has not articulated any basis for arguing that this limitation is satisfied, the Court should find this to be an exceptional case.

**4. DataTern’s Allegations that the Accused Products Include a “Code Generator” Are Baseless**

Claim 10 of the ’502 patent requires a “code generator that employs said map to create at least one interface object.”<sup>7</sup> DataTern has never had any basis for contending that the accused products include a code generator to create an interface object. Indeed, DataTern concedes, in its proposed judgment, that the accused product “does not generate code for a class and instantiate an object from that class.” (Ex. A at ¶ 21.) Because DataTern must have known, when it filed suit, that the accused products do not include a “code generator . . . to create at least one interface object,” the Court should declare this an exceptional case and award MicroStrategy at least its fees that relate to defending against this claim.

---

<sup>7</sup> Claim 1 likewise requires creating an interface object, which the New York court construed to require generating code for the interface object. Claim 10 makes the requirement of generating code explicit in the claim language.

**C. MicroStrategy Should at Least Be Awarded Attorney's Fees Resulting from DataTern's Vexatious Litigation Tactics**

Aside from DataTern's institution of this litigation with no basis for doing so, DataTern has engaged in at least two forms of litigation conduct that justify a finding of exceptional case. *See MarcTec*, 664 F. 3d at 916 ("A case may be deemed exceptional under § 285 where there has been . . . vexatious or unjustified litigation."); *see also Eon-Net LP v. Flagstar Bancorp*, 653 F.3d 1314, 1324 (Fed. Cir. 2011).

**1. DataTern's Lawsuit Against MicroStrategy's Customers Was Motivated By an Improper Purpose**

DataTern's decision to sue MicroStrategy's customers was nothing more than (a) an unwarranted litigation tactic designed to put pressure on MicroStrategy by impairing its relationships with its customers,<sup>8</sup> and (b) an attempt to engage in naked judge-shopping and avoid this District's system of random judicial assignment mandated by Local Rule 40.1(A)(3).

As explained earlier (§ III.A), DataTern filed seventeen lawsuits against MicroStrategy's customers and business partners in November 2011. Shortly thereafter, it dismissed eleven cases not assigned to Judge Stearns. DataTern then filed five additional lawsuits, including the one against MicroStrategy, marking them as related to one of the cases assigned to Judge Stearns—but, in violation of Local Rule 40.1(G)(2), not marking them as related to the earlier, still-pending case against Blazent, assigned to Judge Dein.

Aside from using its customer cases to seek to select the presiding judge, DataTern also used these cases to increase MicroStrategy's costs and business risk. As explained above (§ III.A), DataTern would not say whether or not it had independent claims against the

---

<sup>8</sup> *See Ricoh Co., Ltd. v. Aeroflex Inc.*, 279 F.Supp.2d 554, 557 (D. Del.2003) ("[A] patentee's election to sue customers, rather than the manufacturer itself, is often based on a desire to intimidate smaller businesses.")

customer defendants until April of 2012. In April 2012, DataTern served its infringement contentions, demonstrating that it never had a basis for independent claims against the customers. Then, when the customers then moved to stay the cases against them, DataTern unreasonably opposed, still hoping to increase their costs and its own leverage against MicroStrategy.

Because DataTern's assertions against MicroStrategy's customers were motivated by bad faith—both to evade this Court's random judicial assignment and to increase pressure on MicroStrategy—the Court should declare this to be an exceptional case, and should award MicroStrategy its fees attributable to the lawsuits against the other defendants.

## **2. DataTern Unreasonably Forced MicroStrategy to Collect and Make Available Source Code DataTern Never Intended to Review**

MicroStrategy should be compensated for its attorneys' fees unnecessarily spent gathering, reviewing, and producing source code that DataTern demanded, but never actually intended to review.

DataTern repeatedly excused its own failure to articulate adequate infringement theories by claiming that DataTern was awaiting MicroStrategy's source code. (*See supra*, § III.A.) MicroStrategy obliged, and incurred significant legal fees collecting, reviewing, and preparing its source code for DataTern's review and produced the code on August 10, 2012. Yet despite its repeated requests for the code and promises to supplement its infringement contentions following MicroStrategy's production, in the end, DataTern never reviewed MicroStrategy's source code:

May 2012: “Consistent with the Joint Statement Pursuant to Local Rules 16.1(D) and 16.6, nonetheless, ***DataTern intends to supplement its infringement contentions within 90 days of production of MicroStrategy's source code.***” – (SOF ¶ 20.)

Sept. 2012: “***We disagree with your contention that DataTern is required to inspect the MicroStrategy source code within 90 days.***” – DataTern Counsel Kelly (SOF ¶ 31.)

January 2013 (To the Court): “***DataTern is under no obligation to amend its infringement contentions so that they specifically identify the source code*** that satisfies each claim limitation prior to claim construction.” (D.I. 103 at 3.)

DataTern’s insistence on the production of source code to permit it to provide infringement contentions, followed by its refusal to review the code or to provide the promised contentions, served only to increase MicroStrategy’s litigation costs and to underscore the baseless nature of its infringement allegations. The Court should declare this to be an exceptional case, and DataTern should be required to reimburse MicroStrategy for expenses incurred as a result of DataTern’s vexatious multiplication of costs. *Old Reliable Wholesale, Inc. v. Cornell Corp.*, 635 F.3d 539 (Fed. Cir. 2011) (“Where a party engages in litigation misconduct, fees can be awarded pursuant to section 285...”).

The Declaration of Benjamin K. Thompson, filed herewith, summarizes MicroStrategy’s legal fees incurred to date owing to DataTern’s litigation misconduct. Pursuant to Fed. R. Civ. P. 54(d)(2)(B)(iii), MicroStrategy provides an estimate of the amount sought herein of approximately \$877,000 in fees and costs incurred as a result of DataTern’s assertion of baseless infringement allegations.<sup>9</sup> MicroStrategy will provide full itemized accounting of its fees and costs as appropriate and requested by the Court. *See* Fed. R. Civ. P. 54 (1993 advisory committee notes, Subsection (d), Paragraph (2)) (“The rule does not require that the motion be supported at the time of filing with the evidentiary material bearing on the fees. This material must of course be submitted in due course, according to such schedule as the court may direct in light of the circumstances of the case. What is required is the filing of a motion sufficient to alert the adversary and the court that there is a claim for fees, and the amount of such fees (or a fair estimate).”)

---

<sup>9</sup> MicroStrategy’s estimate is based on services rendered between November 2011 and the end of November 2012. MicroStrategy reserves the right to supplement this amount with additional fees and costs, including for the months of December 2012, January 2013, and February 13, once invoices have been finalized and submitted to MicroStrategy.

## VII. CONCLUSION

For reasons set forth herein, MicroStrategy respectfully requests the Court enter MicroStrategy's proposed judgment, which is filed herewith. Further, MicroStrategy requests this court grant MicroStrategy's motion for attorneys' fees, and enter an appropriate schedule under which the parties would agree or the Court would determine the reasonable amount of such fees.

Dated: February 4, 2013

/s/Thomas A. Brown

Gregory A. Madera (BBO #313020)

madera@fr.com

Thomas A. Brown (BBO #657715)

tbrown@fr.com

Maureen M. Brenner (BBO # 679573)

mbrenner@fr.com

Sivananda K. Reddy (BBO #678598)

reddy@fr.com

Fish & Richardson, P.C.

One Marina Park Drive

Boston, MA 02110-1878

617-542-5070

617-542-8906 (fax)

Benjamin K. Thompson

(admitted *pro hac vice*)

bthompson@fr.com

Fish & Richardson, P.C.

1180 Peachtree St. NE

Atlanta, GA 30318

404-724-2844

Attorneys for Defendant

MICROSTRATEGY INCORPORATED.

**CERTIFICATE OF SERVICE**

I hereby certify that this document(s) filed through the ECF system will be sent electronically to the registered participants as identified on the Notice of Electronic Filing (NEF) and paper copies will be sent to those indicated as non-registered participants on this 4<sup>th</sup> day of February 2013.

/s/Thomas A. Brown

Thomas A. Brown